

**This Page Is Inserted by IFW Operations  
and is not a part of the Official Record**

## **BEST AVAILABLE IMAGES**

**Defective images within this document are accurate representations of the original documents submitted by the applicant.**

**Defects in the images may include (but are not limited to):**

- **BLACK BORDERS**
- **TEXT CUT OFF AT TOP, BOTTOM OR SIDES**
- **FADED TEXT**
- **ILLEGIBLE TEXT**
- **SKEWED/SLANTED IMAGES**
- **COLORED PHOTOS**
- **BLACK OR VERY BLACK AND WHITE DARK PHOTOS**
- **GRAY SCALE DOCUMENTS**

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**

Japanese Patent Laid-open Publication No. HEI 6-222920 A

Publication date : August 12, 1994

Applicant : PHILIPS ELECTRONICS NEAMLOZE VENNOOTSHAP

Title : Data Processor Having Computing Units Sharing a Group of

5 Register Files

[Summary]

[Object] To provide a processor which is more compact and efficient as compared to that based on the conventional technology and also which is based on the considerations to functional mutual  
10 dependency between execution units.

[Configuration] A data processor based on the modular architecture has a plurality of computing units, each of which  
15 is useful for realization of a particular function demanded with a set of instructions deciding operations of the processor. Some of the computing units share a register file, and other computing units share another register file, or have a discrete register file not shared by other computing units. Register files shared  
20 by two or more computing units can physically and functionally be integrated with each other in application thereof. A small register file control circuit and a simple microcode language can be obtained by sacrificing only limitless increase of instruction cycles and a small register file area of the realized IC.

(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開 号

特開平6-222920

(43)公開日 平成6年(1994)8月12日

(51)IntCl<sup>8</sup>

G 0 6 F 9/38

識別記号

3 7 0 B 9193-5B

3 1 0 X 9193-5B

9188-5B

F I

技術表示箇所

7/00

G 0 6 F 7/00

R

審査請求 未請求 請求項の数5 OL (全 6 頁)

(21)出願番号 特願平5-281000

(22)出願日 平成5年(1993)10月19日

(31)優先権主張番号 9 2 2 0 3 1 9 2 : 7

(32)優先日 1992年10月19日

(33)優先権主張国 オランダ (NL)

(71)出願人 582098322

フィリップス エレクトロニクス ネムロ  
ーゼ フェンノートシャップPHILIPS ELECTRONICS  
NEAMLOZE VENNOOTSH  
APオランダ国 5621 ベーアー アイندر  
フェン フルーネヴァウツウェッハ 1

(72)発明者 ヨセフ ルイス パン メールベルゲン

オランダ国 5621 ベーアー アイندر  
フェン フルーネヴァウツウェッハ 1

(74)代理人 弁理士 杉村 暁秀 (外5名)

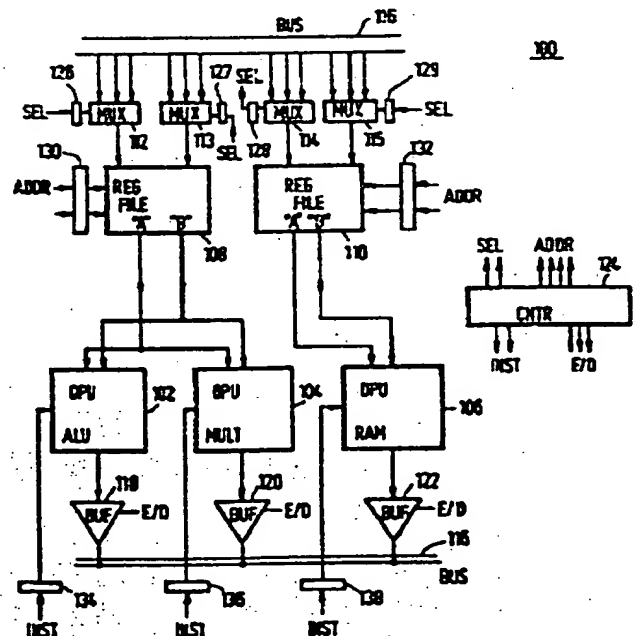
最終頁に続く

(54)【発明の名称】 レジスタファイルのグループを共有する演算ユニットを持つデータプロセッサ

(57)【要約】

【目的】 実行ユニット間の機能的な相互依存性を考慮した、従前のものより小型かつ効率的なプロセッサを与えることを目的とする。

【構成】 モジュラーアーキテクチャのデータプロセッサは複数の演算ユニットを具え、その各々はプロセッサの動作を決定する命令セットにより要求された特定機能の実現に役立っている。いくつかの演算ユニットは同じレジスタファイルを共有し、他の演算ユニットは他のレジスタファイルを共有するか、あるいは他の演算ユニットにより共有されない個別レジスタファイルを有している。2つ以上の演算ユニットのレジスタファイルはその適用に依存して物理的かつ機能的に互いに併合できる。命令サイクルの限界的な増大、具体化されたICの小さいレジスタファイル面積のみを犠牲にして、少ないレジスタファイル制御回路と簡単なマイクロコード語が得られる。



## 【特許請求の範囲】

【請求項1】 第1の複数の演算ユニットを持つデータ処理手段であって、その各々が各データを受信し、かつ各データに基づいて各データ処理演算を実行するもの、データ処理手段に連結され、かつ該データ処理手段にデータを供給する、先立って蓄積データに作用するレジスタファイル手段であって、該レジスタファイル手段が第2の複数のレジスタファイルを具えるもの、およびここで各特定の演算ユニットが特定のレジスタファイルに連結されるもの、を具えるデータプロセッサにおいて、少なくとも1つのレジスタファイルが少なくとも2つの演算ユニットに連結されることを特徴とするデータプロセッサ。

【請求項2】 第1の複数の演算ユニットが相互に非一様であることを特徴とする請求項1に記載のプロセッサ。

【請求項3】 第1レジスタファイルが第1演算ユニットと第2演算ユニットに連結され、ここで第1演算ユニットがダイアディック演算の実行を行い、かつ第2演算ユニットがモナディック演算の実行を行うことを特徴とする請求項1に記載のプロセッサ。

【請求項4】 第1演算ユニットがメモリを具え、かつ第2演算ユニットがアドレス計算ユニットを具えることを特徴とする請求項3に記載のプロセッサ。

【請求項5】 集積回路チップで実現された請求項1のプロセッサ。

## 【発明の詳細な説明】

## 【0001】

【技術分野】 本発明は、第1の複数の演算ユニットを具えるデータプロセッサに関連し、その各々が各データを受信し、かつ各データに基づいて各データ処理演算を実行する。このデータプロセッサは、データ処理手段に連結され、かつ該データ処理手段にデータを供給するのに先立って蓄積データに作用するレジスタファイル手段をさらに具え、該レジスタファイル手段は第2の複数のレジスタファイルを具えている。各特定の演算ユニットは特定のレジスタファイルに連結されている。

## 【0002】

【背景技術】 データプロセッサはたいいていの計算機の心臓部を形成する既知のデバイスである。データプロセッサは予め規定されたアルゴリズムの実行により出力データを創成するよう入力データに演算を行う。典型的には、プロセッサは以下の1つ以上の演算ユニット：加算、減算、シフトを実行する算術演算ユニット（例えばALU）、2つのデータ項目を乗算する乗算器（MUL）、データの読み書きを行うランダムアクセスメモリ（RAM）、読み取りのみを実行する読取専用メモリ（ROM）、メモリをアクセスするアドレスを与えるアドレス計算ユニット、もしくは特定アプリケーションユニット（ASU：application-specific unit）を機能

的に含んでいる。ASUは専用ユニットで、典型的には他の演算ユニットに規定された命令サイクルの限定された数以内で他の演算ユニットの命令セットで適合しない演算の実行に使用された。ASUに余計な演算をマップすることは、他の演算ユニットの命令サイクルの数を増大するか、あるいは他の演算ユニットと同じ種類の付加的演算ユニットを備えることより効率的である。

【0003】冒頭の記事に導入されたようなプロセッサは、アール・ウッズマ（R. Woodsma）、エフ・ビー・エム・ビーンカー（F. P. M. Beenker）ジェー・エル・バンメルベルゲン（J. L. Van Meerbergen）およびシー・ニーセン（C. Niessen）の論文、「ピラミッド：複合DSP適用に対するアーキテクチャ駆動シリコンコンパイラ（PIRAMID：An Architecture-Driven Silicon Compiler for Complex DSP Applications）」、回路とシステムの1990年アイイーイーイー国際シンポジウム（1990 IEEE International Symposium on Circuits and Systems）、ニューオルレアン（New Orleans）、ルイジアナ（Louisiana）州、米国、1990、5、1-3、頁2596-2600に開示されている。この従前の文書の図2と図3はプロセッサのモジュラーアーキテクチャを示している。このアーキテクチャはプロセッサの機能的適用に基づいてシリコンコンパイラ（PIRAMID）により自動的に発生される。シリコンに組み込まれた通常のデータプロセッサとは違って、それは汎用的に構成され、PIRAMIDにより発生されたプロセッサは直面している適用に最適化される。

【0004】前記の従前の技術文書の図2は、データバスにより相互接続され、かつコントローラに蓄積されたマイクロコードプログラムを介して制御された複数の実行ユニットを有するプロセッサのモジュラーアーキテクチャを例示している。実行ユニットの一般構造は前記文書の図3に示されている。実行ユニットはアルゴリズムを実現するのに必要な1つ以上の演算を実行する機能的かつ物理的実体（entity）である。従って、各実行ユニットは実際に演算を実行する上記の種類の演算ユニットと、演算ユニットによりオペランドとして使用された入力データを蓄積し、かつ必要な場合に演算ユニットに入力データを供給するレジスタファイルを具えている。

【0005】一般に、レジスタファイルは「A」レジスタファイルセクションと「B」レジスタファイルセクションを具えている。「A」レジスタファイルセクションは「A」オペランドを供給し、「B」レジスタファイルセクションは演算ユニットに「B」オペランドを供給する。演算ユニットはそれに基づいて双方のオペランドに演算する。例えば、演算ユニットは「A」および「B」オペランドの双方を論理演算もしくは算術演算に従うべきデータとして解釈しよう。もし演算ユニットがRAMを含むなら、それは「A」オペランドを「B」オペランドにより規定されたアドレスに蓄積すべきデータとして

3

受信する。「A」および「B」オペランドの双方を含む演算は「ダイアディック (dyadic)」演算と呼ばれ、一方、単一オペランドのみを含む演算は「モナディック (monadic)」と呼ばれる。後者の例として、ROMを含む演算ユニットは単一オペランド、すなわちROMのアドレスの…を必要とする。従って、ROMレジスタファイルは「A」レジスタファイルセクションを具えるのみである。

【0006】レジスタファイルは単一命令サイクルの読み取り演算と書き込み演算の実行を行う二重ポートデバイス (dual-ported device) である。レジスタファイルは演算ユニットに接続された高速前景メモリ (fast foreground memory) として役立つ。背景メモリ (background memory) は単一ポートデバイス (single-ported device) であり、かつ1つの読み取り演算あるいは1つの書き込み演算を単一命令サイクルで実行できる。従って、背景メモリは前景メモリと比較すると遅く、かつ個別実行ユニット (例えばRAM、ROM) の演算ユニットとして取り扱われる。レジスタファイルは命令レジスタを介してコントローラによりアドレスされ、一方、背景メモリは他の実行ユニットを実現するアドレス計算ユニットを介してアドレスされる。各個別実行ユニットがそれ自身のレジスタファイルを有することに注意されたい。

#### 【0007】

【発明の目的】たとえPIRAMIDアプローチが大きくプロセッサアーキテクチャを最適化しても、モジュール性 (modularity) は相対的に大規模なプロセッサシステムと、プロセッサ能力の相対的に非効率な使用となろう。このことは、とりわけ、実行ユニットが直面する特定の適用の実行ユニットの間で機能的相互依存性を考慮することなく個別に最適化されることによる。従って、本発明の目的は従前の技術のものより小型、かつ、より効率的であり、かつ従前の技術のプロセッサと少なくとも同じ能力を有する冒頭の記事に述べられた種類のデータプロセッサを与えることである。

#### 【0008】

【発明の開示】この目標を達成するために、本発明は第1の複数の演算ユニットを具えるデータ処理手段を与え、その各々が各データを受信し、かつ各データに基づいて各データ処理演算を実行する。データプロセッサは、データ処理手段に連結され、かつ該データ処理手段にデータを供給するに先立って蓄積データに作用するレジスタファイル手段をさらに具えている。該レジスタファイル手段は第2の複数のレジスタファイルを具えている。特定の各演算ユニットは特定のレジスタファイルに連結されている。本発明によると、少なくとも1つのレジスタファイルは少なくとも2つの演算ユニットに連結されている。

【0009】本発明において、いくつかの演算ユニット

4

は同じレジスタファイルを共有し、一方、他の演算ユニットは他のレジスタファイルを共有するか、あるいは他の演算ユニットにより共有されない個別レジスタファイルを有している。本発明は、演算ユニットのレジスタファイルが関連演算ユニットに位置内容 (すなわちオペランド) を供給することによりしばしば空きレジスタ位置を有するという洞察に基づき、かつ同じオペランドが同時にあるいは時間的に分散していくつかの演算ユニットで使用できるという洞察に基づいている。従って、2つ以上の演算ユニットのレジスタファイルは、プロセッサの適用に依存して、物理的かつ機能的に互いに併合できる。

【0010】例えば、繰返し演算において、第1演算ユニットと第2演算ユニットは共有レジスタファイルから同じオペランドを受信する。第1ユニットは第3演算ユニットとなる処理を供給するために所定の態様でオペランドを処理し、一方、第2演算ユニットは次の繰返しで第1ユニットに供給すべき引き続くオペランドを受信オペランドに基づいて計算する。

【0011】利点は多岐にわたっている。共有レジスタファイルは個別に用いられたレジスタファイルの集合より小さい。共有レジスタ位置の使用は実際に要求されたレジスタ蓄積能力を低減する。命令レジスタのようなレジスタの1/Oを機能的に実現する周辺回路もまた共有され、付加的な寸法の低減となる。

【0012】併合されたレジスタに対してアドレス空間が増大されるから、併合されたレジスタの位置のアドレスに関連するマイクロコードのデータフィールドは元来長い。しかし単一アドレスデータフィールドは従前の技術の2つ以上のアドレスデータフィールドの代わりとなる。それ故、マイクロコード語の幅は低減され、かつ小さいコントローラが使用できる。マイクロコード命令の再スケジューリングは必要であり、命令セットの僅かな拡張もまた必要であろう。しかし、これは得られた利点の観点でやりがいがある。

【0013】さらに、複数の小さい回路ブロックは同じ機能を持つ単一の大規模回路ブロックよりも集積回路の具体化で大きな基板面積を一般に要求する。これは回路ブロックの要素が典型的には、使用されたIC技術で安全性マージンと解像度の観点で回路ブロックに保留されるべき基板面積を完全に占有しないという事実による。その上、計算機援用設計環境において、チップに使用する回路ビルディングブロックの配置はパラメータ形式の規格に基づいて自動的に発生される。典型的には、ビルディングブロックの発生された配置は後で追加すべき簡単な相互接続導線の観点でチップ面積に保留すべき長方形から構成されている。チップの占有されていない空間の部分は長方形の数の増大に対して増大する。さらに、配置の自動発生は特定のパラメータ値、あるいはパラメータ値の範囲 (例えば、メモリサイズもしくはメモリ容

5

量)に対して最適化できる。最適化された範囲を越えると、関連配置はチップの面積を余り効率的に使用しないであろう。従って単一ブロックあるいは僅かなブロック内の複数のブロックの同じ機能の併合はそれに比例して非占有面積の大きさを低減する。

【0014】本発明のプロセッサにおいて、第1の複数の演算ユニットは非一様(non-uniform)であろう。すなわち、演算ユニットは相互に異なる機能を有し、従って、非一様アーキテクチャであろう。異なる演算ユニットの間でレジスタファイルをグループ毎に共有することは全体としてプロセッサにとって有利である。

【0015】本発明の別の実施例において、第1のレジスタファイルは第1および第2の演算ユニットに連結され、第1演算ユニットはダイアディック演算の実行を行い、かつ第2演算ユニットはモノディック演算の実行を行う。典型的には、第1演算ユニットはメモリ(例えばRAM)を具え、第2演算ユニットは、例えばアドレスが繰返し計算すべき場合に有利であるアドレス計算ユニット(ACU: address calculating unit)を具えている。

【0016】上述のPIRAMIDプロセッサのようなモジュラーアーキテクチャを有するもの以外に、通常のプロセッサは結合された意図されたすべての機能に必要以上に大きい単一レジスタファイルを具えている。しかし、本発明は専用プロセッサに関連し、ここでモジュラーアプローチは寸法、速度およびコストを最適化するために局部的に僅かに修正される。特に集積回路の具体化において、本発明は魅力的な特徴を与える。

【0017】

【実施例】添付図面を参照して本発明を実例により説明する。

プロセッサアーキテクチャ

図1はプロセッサ10のアーキテクチャの構成例を与える。プロセッサ10は、とりわけ、実行ユニットの間でデータ通信を行う1つ以上のデータバス20を介して相互接続されている実行ユニット12、14、16および18を含んでおり、その動作はコントローラ22により制御されている。コントローラ22は、分岐のようなデータ依存処理を可能にするために、ステータスバス26を介して実行ユニット12と14から受信されたフラグのようなステータス情報に多分依存して、命令バス24を介して実行ユニット12-18に命令を供給する。実行ユニット12は算術演算ユニット機能(ALU)を実現する。実行ユニット14はRAMを含む実行ユニット16と、ROMを含む実行ユニット18のような、メモリの最適化アドレスシーケンスを発生するためのアドレス計算ユニット(ACU)を具えている。

【0018】従前の実行ユニット

図2は従前の技術の典型的な実行ユニット40のブロック

6

線図を示している。実行ユニット40は適用の際に乗算するマルチプレクサ42と44をその入力に備えている。マルチプレクサ42と44は「A」オペランドセクション48と「B」オペランドセクション50を含むレジスタファイル46にバス20のデータライン(示されていない)を選択的に連結する。マルチプレクサ42と44は、命令レジスタ52と54を介して適当な選択信号SELを供給するコントローラ22の制御の下に動作する。レジスタファイル46はマルチプレクサ42と44から受信されたデータを蓄積し、かつオペランドとして受信する演算ユニット56にデータを転送する。セクション48と50は命令レジスタ58と60を介してコントローラ22から読み書きアドレス(ADDR)を受信する。演算ユニット56は命令レジスタ62を介してコントローラ22から受信された命令の制御の下でレジスタファイル46から受信されたデータに演算を実行する。この演算の結果はバッファ64を介してデータバス20に供給され、すなわち命令レジスタ66を介してコントローラ22により選択的にエネーブルあるいはディスエーブル(E/D)される。ROM機能が実行ユニット40で実現される場合、マルチプレクサ44、セクション50および命令レジスタ54と60は存在しない。

【0019】本発明のビルディングブロック

図3は図1のプロセッサ10のアーキテクチャを実現する本発明によるビルディングブロックを図式的に例示している。ブロック100はレジスタファイル108と110から入力データを受信する演算ユニット102、104および106を具えている。レジスタファイル108はマルチプレクサ112と113を介して入力データを受信し、レジスタファイル110はマルチプレクサ114と115を介して入力データを受信する。マルチプレクサ112-115はバス116に接続されている。演算ユニット102、104および106により与えられた結果はバッファ118、120および122それぞれを介してバス116に供給される。マルチプレクサ112-115は選択信号SELを通して命令レジスタ126、127、128および129を介してコントローラ124により制御される。アドレス(ADDR)は命令レジスタ130と132を介してコントローラ124によりレジスタファイル108と110に供給される。演算ユニット102、104および106の動作モードは命令レジスタ134、136および138を介して命令(INST)を供給するコントローラ124により決定される。バッファ118、120および122はコントローラ124を介して選択的にエネーブルおよびディスエーブル(E/D)される。

【0020】図3のブロックは同じ種類の、あるいは図2に示された種類の他の実行ユニットと協働する高いレベルの実行ユニットとして使用するよう適応されていることに注意されたい。図面全体で、単一ラインがデータあるいは他の信号の入力を示すため、あるいは相互接続を示すために描かれており、これらのラインはいくつかの単一導線幅のバスを表すことにも注意されたい。

7

【0021】示された事例では、レジスタファイル108はコントローラ124の制御の下で演算ユニット102と104にオペランドを選択的に与えるのに使用され、一方、レジスタファイル110は演算ユニット106にオペランドを供給する。レジスタファイル108は2つ以上の演算ユニット(102, 104および106のみが示されている)に各オペランドを供給するために、例えば2つ以上の並列に配列された単一入力レジスタファイル(示されていない)を具えることは当業者にとって明らかであろう。一般に別のマルチプレクサ(示されていない)が一方では少なくともいくつかのレジスタファイルの出力と、オペランドの適当な分布をスケジュールするために他方ではいくつかの演算ユニットの入力の間に挿入できることもまた明らかであろう。

【0022】演算ユニット102-106はこの例のダイアディック命令を実行するものと仮定されている。従前の技術と比較すると、演算ユニット102と104のレジスタファイルは単一レジスタファイルに併合されている。

#### 【0023】典型的実例

図4は共用レジスタファイルの典型的な実例の本質のみを例示している。制御構成とI/O構成は図3のものと類似していると仮定されている。RAM180はレジスタファイル182を介して受信されたアドレスでレジスタファイル182を介して受信されたデータを蓄積し、かつバス116にデータを与える。アドレス計算ユニット(ACU)184はRAM180が次のアドレスを繰返し計算するのと同じアドレスを受信する。すると次のアドレスはバス116に供給され、かつ再びレジスタファイル182に経路される。図4の中断された相互接続はここでは簡単のために省略された制御とI/O構成の残りとして使用されている。

【0024】いくつかの演算ユニットのレジスタファイルが共に併合できるかどうかは、実行すべき実際の演算に依存する。どんな態様で命令セットがレジスタファイルの併合に影響するかどうかは以下に説明される。併合戦略を達成するために、設計者は基本としてどの演算ユニットがどの命令サイクルで能動であることを示すために演算ユニットの初期スケジュールを取るであろう。同時に作動しない、すなわち同じ命令サイクルの間に、演算ユニットのレジスタファイルは併合できる。しかし、そのような命令は希である。もっとありそうなことは、僅かな命令サイクルのみが同時に作動する演算ユニットが存在しよう。レジスタファイルの併合は達成できるが、しかし衝突を回避するために追加の命令サイクルの導入を犠牲にしている。命令サイクルの僅かに増大した数は併合されたレジスタファイルの技術的利点に引き合うであろう。設計者は同時であるか、あるいは異なる命令サイクルのいずれかで1つ以上の演算ユニットで使用されたオペランドを考慮できよう。そのようなオペランドは単一レジスタ位置のみを要求し、それは命令サイクルの

8

延長された数を多分占有するであろう。ハードウェアの節約が、本発明のプロセッサがその実行に対して創成すべきプログラムに主として依存するであろうことは当業者にとって明らかであろう。本発明のプロセッサの集積回路の具体例に対して、基板面積の節約が典型的に15%-20%程度であることが見いだされた。

#### 【図面の簡単な説明】

【図1】図1はデータプロセッサのモジュラーアーキテクチャの一例を例示している。

10 【図2】図2は図1の構造を実現する従前のビルディングブロックを与える。

【図3】図3は図1のデータプロセッサを実現する本発明のビルディングブロックの一例を示している。

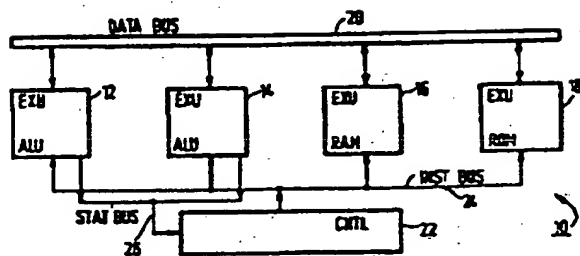
【図4】図4は同じレジスタファイルを共有する演算ユニットの典型的な組合せを開示している。

#### 【符号の説明】

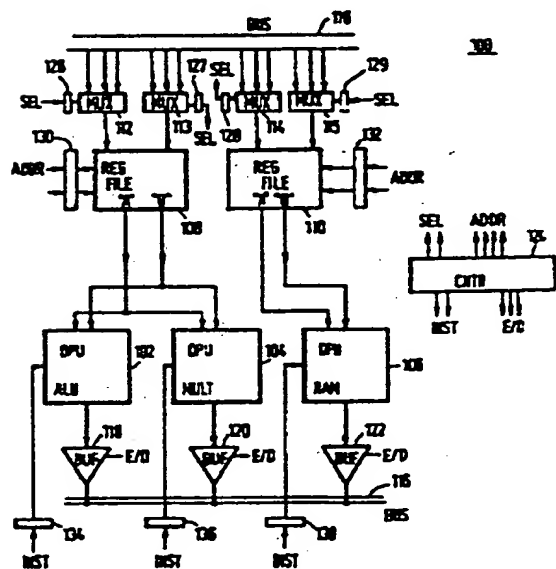
- 10 プロセッサ
- 12 実行ユニット
- 14 実行ユニット
- 20 実行ユニット
- 18 実行ユニット
- 20 データバス
- 22 コントローラ
- 24 命令バス
- 26 ステータスバス
- 40 実行ユニット
- 42 マルチプレクサ
- 44 マルチプレクサ
- 46 レジスタファイル
- 30 48 「A」オペランドセクション
- 50 「B」オペランドセクション
- 52 命令レジスタ
- 54 命令レジスタ
- 56 演算ユニット
- 58 命令レジスタ
- 60 命令レジスタ
- 62 命令レジスタ
- 64 バッファ
- 66 命令レジスタ
- 40 100 ブロック
- 102 演算ユニット
- 104 演算ユニット
- 106 演算ユニット
- 108 レジスタファイル
- 110 レジスタファイル
- 112 マルチプレクサ
- 113 マルチプレクサ
- 114 マルチプレクサ
- 115 マルチプレクサ
- 50 116 バス

118 バッファ  
120 バッファ  
122 バッファ  
124 コントローラ  
126 命令レジスタ  
127 命令レジスタ  
128 命令レジスタ  
129 命令レジスタ

【図1】

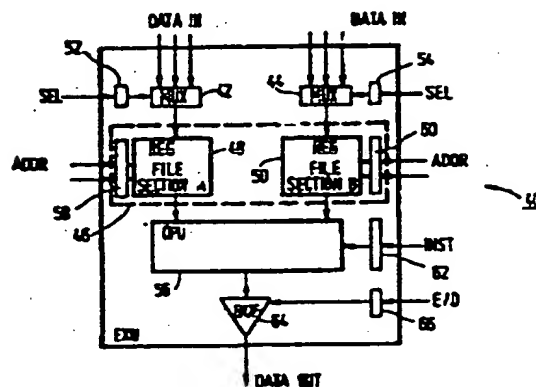


【図3】

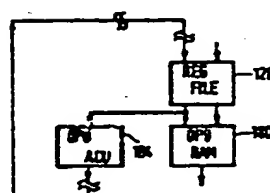


130 命令レジスタ  
132 命令レジスタ  
134 命令レジスタ  
136 命令レジスタ  
138 命令レジスタ  
180 RAM  
182 レジスタファイル  
184 アドレス計算ユニット (ACU)

【図2】



【図4】



フロントページの続き

(72) 発明者 ヘンドリカス アンドリース ヒルデリンク  
オランダ国 5621 ベーアー アイन्दー  
フェン フルーネヴァウツウェッハ 1

(72) 発明者 パウル オイゲン リチャード リッペン  
ス  
オランダ国 5621 ベーアー アイन्दー  
フェン フルーネヴァウツウェッハ 1

(72) 発明者 アントイン デラルエーレ  
オランダ国 5621 ベーアー アイन्दー  
フェン フルーネヴァウツウェッハ 1